What is claimed is:

1. A decoding method comprising:

receiving an encoded signal;

demodulating the received encoded signal to produce soft information; and

iteratively processing the soft information with one or more soft-in / soft-output (SISO) modules, at least one SISO module using a tree structure to compute forward and backward state metrics.

2. The method of claim 1 wherein the at least one SISO computes the forward and backward state metrics by performing recursive marginalization-combining operations.

3. The method of claim 2 wherein the recursive marginalization-combining operations comprise min-sum operations.

4. The method of claim 2 wherein the recursive marginalization-combining operations comprise min*-sum operations.

5. The method of claim 4 wherein $\min^* = \min(x,y) - \ln(1 + e^{-|x-y|})$.

6. The method of claim 2 wherein the recursive marginalization-combining operations comprise sum-product operations.

7. The method of claim 2 wherein the recursive marginalization-combining operations comprise max-product operations.

8. The method of claim 1 wherein the encoded signal comprises at least one of a turbo encoded signal, a block turbo encoded signal, a low density parity check coded signal, a product coded signal, and convolutional coded signal.

9. The method of claim 1 wherein the encoded signal comprises at least one of a parallel concatenated convolutional code and a serial concatenated convolutional code.

10. The method of claim 1 further comprising using the iterative decoding method in a wireless communications system.

11. The method of claim 1 further comprising terminating the iterative processing upon occurrence of a predetermined condition.

12. The method of claim 1 wherein the iterative processing comprises performing parallel prefix operation or parallel suffix operations, or both, on the soft information.

13. The method of claim 1 wherein the iterative processing comprises using soft output of a first SISO as soft input to another SISO.

14. The method of claim 1 wherein the tree structure used by at least one SISO comprises a tree structure that results in the SISO having a latency of $O(\log_2 N)$, where N is a block size.

15. The method of claim 1 wherein the tree structure used by at least one SISO comprises a Brent-Kung tree.

16. The method of claim 1 wherein the tree structure used by at least one SISO comprises a forward-backward tree.

17. The method of claim 16 wherein the forward-backward tree comprises a tree structure recursion that is bi-directional.

18. The method of claim 1 wherein the processing performed by at least one SISO comprises:

tiling an observation interval into subintervals; and

applying a minimum half-window SISO operation on each subinterval.

19. The method of claim 1 wherein the iterative processing comprises performing marginalization-combining operations which form a semi-ring over the soft-information.

20. A soft-in / soft-out (SISO) module comprising:

a plurality of fusion modules arranged into a tree structure and adapted to compute forward and backward state metrics,

wherein each fusion module is defined by the equation:

$$\mathbf{C}(k_0, k_1) \triangleq \mathbf{C}(k_0, m) \otimes_C \mathbf{C}(m, k_1) \iff C(s_{k_0}, s_{k_1}) = \min_{s_m} [C(s_{k_0}, s_m) + C(s_m, s_{k_1})] \ \forall s_{k_0}, s_{k_1}$$

where $\mathbf{C}(k, m)$ is a matrix of minimum sequence metrics (MSM) of state pairs s_k and s_m based on soft-inputs between s_k and s_m.

21. The SISO module of claim 20 wherein at least one of the fusion modules computes forward and backward state metrics by performing recursive marginalization-combining operations.

22. The SISO module of claim 21 wherein the recursive marginalization-combining operations comprise min-sum operations.

23. The SISO module of claim 21 wherein the recursive marginalization-combining operations comprise min*-sum operations.

24. The SISO module of claim 21 wherein min* = $\min(x,y) - \ln(1 + e^{-|x-y|})$.

25. The SISO module of claim 21 wherein the recursive marginalization-combining operations comprise sum-product operations.

26. The SISO module of claim 21 wherein the recursive marginalization-combining operations comprise max-product operations.

27. A soft-in / soft-out (SISO) module comprising:

one or more complete fusion modules (CFMs) for performing marginalization-combining operations in both a forward direction and a backward direction;

one or more forward fusion modules (fFMs) for performing marginalization-combining operations only in the forward direction; and

one or more backward fusion modules (bFMs) for performing marginalization-combining operations only in the backward direction,

wherein the one or more CFMs, fFMs, and bFMs are arranged into a tree structure.

28. The SISO module of claim 27 wherein an amount of the CFMs is a minimum number needed to compute a soft-inverse.

29. The SISO module of claim 28 wherein fFMs and bFMs are used in the tree structure in place of CFMs wherever possible.

30. The SISO module of claim 27 wherein the marginalization-combining operations performed by one or more of the fusion modules comprise min-sum operations.

31. The SISO module of claim 27 wherein the recursive marginalization-combining operations comprise min*-sum operations.

32. The SISO module of claim 31 wherein $\min^* = \min(x,y) - \ln(1 + e^{-|x-y|})$.

33. The SISO module of claim 27 wherein the recursive marginalization-combining operations comprise sum-product operations.

34. The SISO module of claim 27 wherein the recursive marginalization-combining operations comprise max-product operations.

35. The SISO module of claim 27 wherein the tree structure comprises at least one of a Brent-Kung tree and a forward-backward tree (FBT).

36. A method of iterative detection comprising:

receiving an input signal corresponding to one or more outputs of a finite state machine (FSM); and

determining the soft inverse of the FSM by computing forward and backward state metrics of the received input signal using a tree structure.

37. The method of claim 36 wherein the forward and backward state metrics are computed by at least one soft-in / soft-out (SISO) module.

38. The method of claim 36 wherein the forward and backward state metrics are computed using a tree-structured set of marginalization-combining operations.

39. The method of claim 38 wherein the marginalization-combining operations comprise min-sum operations.

40. The method of claim 38 wherein the marginalization-combining operations comprise min*-sum operations.

41. The method of claim 40 wherein $\min^* = \min(x,y) - \ln(1 + e^{-|x-y|})$.

42. he method of claim 38 wherein the marginalization-combining operations comprise sum-product operations.

43. The method of claim 38 wherein the marginalization-combining operations comprise max-product operations.

44. The method of claim 36 wherein the input signal comprises at least one of a turbo encoded signal and a convolutional coded signal.

45. The method of claim 36 wherein the input signal comprises at least one of a parallel concatenated convolutional

encoded signal and a serial concatenated convolutional encoded signal.

46. The method of claim 36 wherein determining the soft inverse of the FSM comprises iteratively processing soft information.

47. The method of claim 46 wherein the iterative processing comprising performing parallel prefix operation or parallel suffix operations, or both, on the soft information.

48. The method of claim 46 wherein the iterative processing comprises using soft output of a first SISO as soft input to another SISO.

49. The method of claim 37 wherein the tree structure used comprises a tree structure that results in the SISO module having a latency of $O(\log_2 N)$, where N is a block size.

50. The method of claim 36 wherein the tree structure comprises a Brent-Kung tree.

51. The method of claim 36 wherein the tree structure comprises a forward-backward tree.

52. The method of claim 51 wherein the forward-backward tree comprises a tree structure recursion that is bi-directional.

53. The method of claim 37 wherein the at least one SISO further:

tiles an observation interval into subintervals; and

applies a minimum half-window SISO operation on each subinterval.

54. A turbo decoder comprising:

a demodulator adapted to receive as input a signal encoded by a finite state machine (FSM) and to produce soft information relating to the received signal; and

at least one soft-in / soft-out (SISO) module in communication with the demodulator and adapted to compute a soft-inverse of the FSM using a tree structure.

55. The decoder of claim 54 wherein the tree structure implements a combination of parallel prefix and parallel suffix operations.

56. The decoder of claim 54 further comprising at least two SISO modules in communication with each other, wherein the SISO

modules iteratively exchange soft information estimates of the decoded signal.

57. The decoder of claim 54 wherein at least one SISO computes the soft-inverse of the FSM by computing forward and backward state metrics of the received signal.

58. The decoder of claim 54 wherein the tree structure used by at least one SISO comprises a tree structure that results in the SISO having a latency of $O(\log_2 N)$, where N is a block size.

59. The decoder of claim 54 wherein the tree structure used by at least one SISO comprises a Brent-Kung tree.

60. The decoder of claim 54 wherein the tree structure used by at least one SISO comprises a forward-backward tree (FBT).

61. A method of iterative detection comprising:

receiving an input signal corresponding to output from one or more block encoding modules; and

determining the soft inverse of the one or more block encoding modules by computing forward and backward state metrics of the received input signal using a tree structure.

62. The method of claim 61 wherein the forward and backward state metrics are computed by at least one soft-in / soft-out (SISO) module.

63. The method of claim 61 wherein the forward and backward state metrics are computed using a tree-structured set of marginalization-combining operations.

64. The method of claim 63 wherein the marginalization-combining operations comprise min-sum operations.

65. The method of claim 63 wherein the marginalization-combining operations comprise min*-sum operations.

66. The method of claim 65 wherein $\min^* = \min(x,y) - \ln(1 + e^{-|x-y|})$.

67. The method of claim 63 wherein the marginalization-combining operations comprise sum-product operations.

68. The method of claim 63 wherein the marginalization-combining operations comprise max-product operations.

69. The method of claim 63 wherein the input signal comprises at least one of a block turbo encoded signal, a low density parity check coded signal, and a product coded signal.

70. The method of claim 63 wherein determining the soft inverse of the block encoding module comprises iteratively processing soft information.

71. The method of claim 70 wherein the iterative processing comprising performing parallel prefix operation or parallel suffix operations, or both, on the soft information.

72. The method of claim 70 wherein the iterative processing comprises using soft output of a first SISO as soft input to another SISO.

73. The method of claim 62 wherein the tree structure used comprises a tree structure that results in the SISO module having a latency of O(log2 N), where N is a block size.

74. The method of claim 61 wherein the tree structure comprises a Brent-Kung tree.

75. The method of claim 61 wherein the tree structure comprises a forward-backward tree.

76. The method of claim 75 wherein the forward-backward tree comprises a tree structure recursion that is bi-directional.

77. The method of claim 62 wherein the at least one SISO further:

tiles an observation interval into subintervals; and

applies a minimum half-window SISO operation on each subinterval.

78. A block decoder comprising:

a demodulator adapted to receive as input a signal encoded by a block encoding module and to produce soft information relating to the received signal; and

at least one soft-in / soft-out (SISO) module in communication with the demodulator and adapted to compute a soft-inverse of the block encoding module using a tree structure.

79. The decoder of claim 78 wherein the tree structure implements a combination of parallel prefix and parallel suffix operations.

80. The decoder of claim 78 further comprising at least two SISO modules in communication with each other, wherein the SISO

modules iteratively exchange soft information estimates of the decoded signal.

81. The decoder of claim 78 wherein at least one SISO computes the soft-inverse of the block encoding module by computing forward and backward state metrics of the received signal.

82. The decoder of claim 78 wherein the tree structure used by at least one SISO comprises a tree structure that results in the SISO having a latency of O(log2 N), where N is a block size.

83. The decoder of claim 78 wherein the tree structure used by at least one SISO comprises a Brent-Kung tree.

84. The decoder of claim 78 wherein the tree structure used by at least one SISO comprises a forward-backward tree (FBT).

85. An iterative detection method comprising:

receiving an input signal corresponding to one or more outputs of a module whose soft-inverse can be computed by running the forward-backward algorithm on a trellis representation of the module; and

determining the soft inverse of the module by computing forward and backward state metrics of the received input signal using a tree structure.

86. The method of claim 85 wherein the input signal comprises at least one of a block error correction encoded signal, a block turbo encoded signal, a low density parity check coded signal, and a product coded signal.

87. The method of claim 85 wherein the input signal comprises at least one of a turbo encoded signal and convolutional coded signal.

88. The method of claim 85 wherein the encoded signal comprises at least one of a parallel concatenated convolutional code and a serial concatenated convolutional code.

89. The method of claim 85 wherein the module comprises a finite state machine.

90. The method of claim 85 wherein the module comprises a block encoding module.